The Knapsack Problem

The classic Knapsack problem is typically put forth as:

A thief breaks into a store and wants to fill his knapsack with as much value in goods as possible before making his escape. Given the following list of items available, what should he take?

- Item A, weighing w_A pounds and valued at v_A
- Item B, weighing w_B pounds and valued at v_B
- Item C, weighing w_C pounds and valued at v_C

•••

The Knapsack Problem

- Input
  - Capacity K
  - n items with weights w_i and values v_i
- Goal
  - Output a set of items S such that
    - the sum of weights of items in S is at most K
    - and the sum of values of items in S is maximized

Definining subproblems

- Define P(i,w) to be the problem of choosing a set of objects from the first i objects that maximizes value subject to weight constraint of w.
- V(i,w) is the value of this set of items
- Original problem corresponds to V(n, K)

Recurrence Relation

- $V(i,w) = \max(V(i-1,w-w_i) + v_i, V(i-1, w))$
  - A maximal solution for P(i,w) either
    - uses item i (first term in max)
    - or does NOT use item i (second term in max)
- $V(0,w) = 0$ (no items to choose from)
- $V(i,0) = 0$ (no weight allowed)

(a) Solve the following instance of the $\{0,1\}$ Knapsack Problem with four items where the maximum allowed weight is $W_{\max} = 10$.

i	1	2	3	4
b_i	25	15	20	36
w_i	7	2	3	6

	0	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	25	25	25	25
2	0	0	15	15	15	15	15	25	25	40	40
3	0	0	15	20	20	35	35	35	35	40	45
4	0	0	15	20	20	35	36	36	51	56	56